**Universidad Central del Ecuador**

**Carrera de Ingeniería en Informática**

**“Aplicación movil para detección de vehículos con IA Generativa (LLM)”**

**Yoshua Calahorrano**

**John Guerra**

1. ***FASE DE COMPRENSIÓN DEL NEGOCIO***

**Descripción del Negocio:**

Este proyecto tiene como objetivo el desarrollo de un sistema automatizado de reconocimiento de vehículos utilizando inteligencia artificial generativa a través de la API de Gemini, la cual procesará imágenes de vehículos y almacenará información relevante como la marca, modelo, año, precio y reseña, donde se almacenara en una base de datos MySQL 5.7. A través de una estructura de microservicios orquestados con Docker Compose y gestionados con Nginx, el sistema promete mejorar tanto la precisión como la eficiencia en la extracción de datos de los vehículos con un enfoque de disponibilidad en el mercado ecuatoriano.

**1.1 Objetivos del Negocio**

Objetivo General:

Desarrollar e implementar un sistema de reconocimiento de vehículos que utilice la IA generativa de Gemini para analizar imágenes de vehículos, extraer información específica como marca, modelo, año, precio y reseña, y almacenarla en una base de datos MySQL, asegurando resultados rápidos y precisos para usuarios en Ecuador.

Objetivos Específicos:

* Implementar un microservicio para analizar imágenes de vehículos mediante el API de Gemini.
* Desarrollar un sistema para almacenar la información de los vehículos en una base de datos MySQL.
* Integrar los microservicios utilizando Docker Compose para asegurar la escalabilidad y la facilidad de administración de contenedores.
* Desarrollar una aplicación móvil utilizando Flutter para mostrar los resultados de los vehículos analizados.
  1. **Criterios de Éxito**

**Precisión del sistema:** Se espera que el modelo de reconocimiento logre una precisión mayor al 90% en la identificación de vehículos.

**Tiempo de respuesta:** El sistema debe procesar las imágenes y devolver los resultados en menos de 5 segundos.

**Almacenamiento:** Los datos extraídos deben almacenarse correctamente en MySQL sin errores de duplicidad o pérdida.

**Escalabilidad:** El sistema debe ser capaz de manejar múltiples solicitudes simultáneas utilizando contenedores Docker de manera eficiente.

* 1. **Valoración de la Situación**

**1.2.1 Recursos y Requisitos**

Tecnologías:

* Backend: FastAPI (Python)
* Base de Datos: MySQL 5.7
* Orquestación: Docker Compose
* Servidor: Nginx
* Aplicación Móvil: Flutter
* Inteligencia Artificial: Gemini API de Google para el análisis de imágenes de vehículos.

Recursos humanos:

* Equipo de desarrollo backend (Python).
* Equipo de desarrollo frontend (Flutter).
  + 1. **Supuestos y Restricciones**

**Supuestos:**

* Se asumirá que las imágenes de los vehículos serán de calidad suficiente para ser procesadas por el sistema.
* La API de Gemini proporcionará resultados precisos y consistentes durante el análisis de las imágenes.

**Restricciones:**

* El tiempo de procesamiento de cada imagen no debe superar los 10 segundos para garantizar una experiencia de usuario óptima.
* La base de datos debe manejar una cantidad de datos que no supere el límite de almacenamiento para la infraestructura disponible.
* Los resultados se basan en el mercado ecuatoriano, si no se comercializa algún modelo en Ecuador se lo expondrá en la aplicación.

**1.2.3 Riesgos y Contingencias**

**Riesgo:** El modelo de IA basado en Gemini podría no alcanzar la precisión esperada, lo que afectaría la calidad de los resultados.

**Contingencia:** Desarrollar un sistema de evaluación para detectar posibles fallos de precisión y ajustar el modelo o la infraestructura si es necesario.

**Riesgo:** Posibles problemas en la integración entre los microservicios.

**Contingencia:** Implementar pruebas unitarias y de integración para cada microservicio y hacer un seguimiento constante durante el desarrollo.

**1.2.4 Terminología**

**API de Gemini:** Plataforma de inteligencia artificial generativa que permite el análisis y extracción de datos a partir de imágenes.

**Microservicios:** Arquitectura de software donde cada componente realiza una función específica, como el análisis o almacenamiento de datos.

**Docker Compose:** Herramienta para definir y ejecutar aplicaciones multicontenedor en Docker.

**FastAPI:** Framework de Python para crear APIs de alto rendimiento.

**1.2.5 Beneficios**

**Optimización:** Reducción del tiempo de respuesta en el análisis de vehículos.

**Escalabilidad:** Uso de Docker Compose para facilitar la implementación y escalabilidad del sistema.

**Precisión:** Mejor precisión en el reconocimiento de vehículos con el uso de Gemini API.

**1.3 Objetivos de Minería de Datos**

* **Objetivo General:**

Desarrollar un sistema eficaz para el reconocimiento de vehículos mediante la integración de LLM (Gemini API/ IA Generativa), con la finalidad de analizar imágenes y almacenar información detallada sobre vehículos, como la marca, modelo, año, precio estimado y una reseña. Este sistema debe ser escalable, rápido y preciso, y ofrecer un servicio confiable para el mercado ecuatoriano.

* **Objetivos Específicos:**
* Implementar un microservicio de análisis de imágenes de vehículos utilizando la Gemini API.
* Crear un microservicio que almacene los datos de los vehículos en una base de datos MySQL, para garantizar la integridad de los datos.
* Desarrollar un sistema que orqueste los microservicios (análisis, almacenamiento y visualización) utilizando Docker Compose y Nginx como servidor.
* Implementar una aplicación móvil en Flutter que permita consultar los vehículos procesados y visualizar sus detalles en tiempo real.

**1.3.1 Criterios de Éxito de Minería de Datos**

* 1. **Plan del Proyecto**

El proyecto se llevará a cabo en las siguientes fases, que corresponden a las etapas de desarrollo e implementación del sistema, utilizando la metodología CRISP-DM:

|  |  |
| --- | --- |
| FASES | TÉCNICAS |
| * Fase 1: Comprensión del negocio | * Identificar vehículos a partir de imágenes y entregar información útil (marca, modelo, año, precio, reseña) para posibles aplicaciones en movilidad, comercio o catastro vehicular. Usuarios: cualquier persona con móvil Android. |
| * Fase 2: Comprensión de los datos | * Imágenes de vehículos capturadas desde cámara o galería. Se evalúa si contienen vehículos (verify\_phase) y qué tan adecuados son para análisis. Se definen criterios de validez (imagen clara, frontal, buena iluminación). |
| * Fase 3: Preparación de los datos | * Filtro automático con verify\_phase que evita procesar imágenes irrelevantes. Estructuración de prompts (prompt\_builder) y extracción en analyze\_phase. Conversión de resultados en JSON estructurado. |
| * Fase 4: Modelado | * Uso de IA generativa (Gemini 2.0 Flash) como modelo de inferencia. Es un modelo pre entrenado, con prompts y fases (verificación y análisis) que afectan el rendimiento. También definiendo una arquitectura REST (FastAPI) y flujos entre microservicios. |
| * Fase 5: Evaluación del modelo | * Verificación del rendimiento del sistema con datos reales: ¿Gemini responde correctamente? ¿El análisis detecta bien los datos? ¿El historial se guarda correctamente? Se revisa JSON generado y comportamiento en la app móvil. |
| * Fase 6: Implementación | * Despliegue en Docker + NGINX, ejecutando en un ec2 de AWS. Exposición del backend unificador como API REST (/analyze). Integración con aplicacion móvil Flutter mediante ip fija de ec2. Resultados guardados en MySQL. Página web en NGINX y uso de tarjetas de historial. |

Tabla 1. Plan de Proyecto

1. ***FASE DE COMPRENSIÓN DE LOS DATOS***

**2.1 Recolección de datos iniciales**

Los datos provendrán de las imágenes de vehículos proporcionadas por los usuarios a través de la aplicación móvil. Estas imágenes serán procesadas por el microservicio de análisis utilizando la API de Gemini para extraer la información necesaria.

**2.2 Descripción de los datos**

**Diccionario de datos:**

**Los datos extraídos de las imágenes incluirán:**

|  |  |  |
| --- | --- | --- |
| **VARIABLE**  (Columnas) | **DESCRIPCIÓN** | **CONTENIDO**  (Filas) |
| Marca | El fabricante del vehículo | Mazda, Toyota, Audi, etc. |
| Modelo | El modelo específico del vehículo | RX-8, Q8 e-tron, etc. |
| **Año** | El año de fabricación del vehículo | 2020, 2019, 2018, etc. |
| **Precio** | Estimación del precio del vehículo en Ecuador | 8000 USD, 35000 USD, etc. |
| **Reseña** | Descripción breve del vehículo | |  | | --- | |  |  |  | | --- | | "Deportivo japonés", "SUV de lujo" | |

Tabla 2. Detalle de las variables.

**Ejemplo de datos:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ford | F-150 Raptor | 2023 | 95000 | Pickup a gasolina, 3.5L v6 Turbo, Dimensiones grandes para la ciudad. |
| Mazda | RX-8 | 2004 | 15000 | Deportivo a gasolina con motor rotativo de 1.3 litros |
| Bmw | M3 GTR (E46) | 2001 | 150000 | Motor gasolina, cilindraje 3.2L, no disponible en el mercado ecuatoriano. |

**2.3 Verificación de la calidad de los datos**

Se validará la calidad de los datos comparando los resultados generados por Gemini con una base de datos de referencia para garantizar que los datos sean correctos y completos.

**2.4 Exploración de los datos**

La exploración de los datos se realizará al analizar las primeras imágenes procesadas para asegurar que la API esté extrayendo correctamente los datos relevantes (marca, modelo, año, precio, reseña).

1. ***FASE DE PREPARACIÓN DE LOS DATOS***
   1. **Selección de los datos**Las imágenes que se utilizarán deberán cumplir con un mínimo de calidad para asegurar que el sistema las procese correctamente.

**3.2 Limpieza de los datos**

Se eliminarán los registros duplicados en la base de datos y los datos incorrectos como valores nulos o erróneos.

**3.3 Estructuración de los datos**

Los datos se estructurarán en tablas en la base de datos MySQL con campos específicos para cada atributo del vehículo.

**3.4 Integración de los datos**

Los datos extraídos de las imágenes se integrarán automáticamente a la base de datos MySQL mediante un microservicio que manejará el almacenamiento de los resultados.

**3.5 Formateo de los datos**

Los datos serán formateados en un **JSON estándar** para facilitar su uso y visualización en la aplicación móvil.

1. ***FASE DE MODELADO***
   1. **Selección de la Técnica de Modelado**

El modelo de análisis utilizará la API de Gemini, una solución basada en IA generativa entrenada previamente, que ofrece resultados precisos en el análisis de imágenes y la extracción de datos relevantes sobre los vehículos. Esta solución se considera la más eficiente, ya que fue entrenada previamente y ha mostrado mejores resultados en términos de tiempo y precisión que otros modelos como YOLO.

**4.2 Generar el Plan de Pruebas**

Se realizará una serie de pruebas para evaluar la precisión del sistema en el reconocimiento de vehículos y la respuesta generada. También se implementarán pruebas de estrés para verificar la capacidad del sistema en escenarios de alta demanda.

**4.3 Construcción del Modelo**

**4.3.1 Desarrollo del Modelo**

El desarrollo del modelo de reconocimiento de vehículos incluye los siguientes pasos:

* **Entrenamiento de la Gemini API:** Se utilizará la API de Gemini para procesar las imágenes y extraer la información relevante de los vehículos. Esta API ha sido pre entrenada con grandes volúmenes de datos y es capaz de generar una alta precisión.
* **Desarrollo del microservicio de análisis:** Se implementará un microservicio en FastAPI que recibirá las imágenes, las enviará a la API de Gemini para su procesamiento y recibirá los datos generados.
* **Almacenamiento en MySQL:** Los datos procesados serán almacenados en una base de datos MySQL que será gestionada por otro microservicio.
* **Orquestación con Docker Compose:** Se utilizará Docker Compose para gestionar los diferentes microservicios (análisis, almacenamiento, etc.),
* **Implementación de Nginx:** El servidor Nginx se utilizará como un proxy inverso para manejar las solicitudes entrantes hacia los microservicios, garantizando una distribución eficiente de tráfico.
* **Aplicación móvil con Flutter:** Para la visualización de los datos, se desarrollará una aplicación móvil en Flutter, que permitirá a los usuarios consultar los detalles de los vehículos procesados, incluyendo imágenes y especificaciones.

1. ***EVALUACIÓN DEL MODELO***

5.1 Métricas de Evaluación

* Para evaluar la efectividad del modelo, se utilizarán las siguientes métricas:
* Precisión del Reconocimiento: Se medirá la precisión del modelo comparando los datos extraídos por Gemini con una base de datos de referencia.
* Tiempo de Respuesta: Se medirá el tiempo desde que se envía una imagen hasta que se recibe la respuesta con los datos procesados (en segundos).
* Eficiencia en el Almacenamiento: Se verificará la eficacia del microservicio encargado de almacenar los datos, asegurando que no haya errores de duplicidad o pérdida de información.
* Robustez del Sistema: Se evaluará cómo responde el sistema a imágenes de diferentes tipos y calidades.

5.2 Plan de Pruebas

* Pruebas Unitarias: Cada microservicio será probado de manera aislada para verificar que funcione correctamente.
* Pruebas de Integración: Se probará la integración de todos los microservicios para verificar su funcionamiento conjunto.
* Pruebas de Carga: El sistema será sometido a altas cargas de trabajo para medir su capacidad de procesamiento.
* Pruebas de Usabilidad: Se probará la aplicación móvil en diferentes dispositivos para garantizar que los resultados sean fáciles de visualizar.

5.3 Resultados Esperados

* Precisión de al menos el 90% en el reconocimiento de vehículos.
* Tiempo de respuesta menor a 5 segundos por consulta.
* Más del 95% de las imágenes procesadas con éxito.
* Almacenamiento correcto del 100% de los datos en la base de datos.

1. **FASE DE IMPLEMENTACIÓN.**

**6.1 Trabajo Futuro**

En el futuro, se espera integrar funciones como verificación de estado vehicular (multas, impuestos), conexión con APIs públicas de tránsito del ecuador, y análisis de tendencias de precios por marca y modelo. También se podrá implementar un sistema de recomendación personalizado.

**6.2 Conclusiones**

El desarrollo de este sistema de reconocimiento vehicular utilizando IA generativa ha permitido comprobar la viabilidad de aplicar modelos LLM como Gemini en contextos prácticos.

Gracias a su estructura modular, escalable y distribuida, el sistema está listo para escalarse y adaptarse a nuevas funcionalidades orientadas al mercado ecuatoriano.

**6.3 Agradecimientos**

Agradecemos al equipo de desarrollo, a los proveedores de la API de Gemini, y a todos los involucrados en la implementación y pruebas de este sistema, la cual este proyecto ha sido posible gracias a los recursos proporcionados por la Universidad Central del Ecuador, especialmente a los docentes y compañeros de la carrera de Ingeniería en Sistemas.